

"Embedded Linux lernen mit dem Raspberry Pi"

Wenn Sie mit Ihrem Buch einen Raspberry Pi der zweiten Generation (B2) verwenden, dann sollten Sie Folgendes wissen:

Stand: So 1. Mär 15:11:39 CET 2015

Autor: Jürgen Quade

Unterschiede zwischen den Raspberry Pis der ersten Generation (A, B, B+) zum Raspberry Pi B2 bestehen im Bootloader und im Kernel. Ist in der Datei "config.txt" kein Eintrag, der mit "kernel=" beginnt, überprüft der "neue" Bootloader, ob es sich um einen Raspberry Pi der ersten oder der zweiten Generation handelt. Bei einem RPi der ersten Generation wird die Datei `kernel.img` in den Speicher geladen, bei der zweiten Generation `kernel7.img`. Die "7" steht hier stellvertretend für die Prozessorarchitektur. Bei einem B2 handelt es sich um ARMv7, die erste Generation basiert auf ARMv6.

Auch wenn eine ARMv7-CPU Programme abarbeiten kann, die für einen ARMv6 kompiliert wurden, trifft dies auf den Kernel nicht zu.

Für das Buch ergeben sich damit die folgenden Anpassungen:

- Verwenden Sie einen aktuellen Kernel, beispielsweise Version 3.18.8.
- Wählen Sie anstelle von `bcmrpi_defconfig` die Konfiguration `bcm2709_defconfig`

###Seite 66, Beispiel 3.8

Generieren eines Linux-Kernels für den Raspberry Pi B2. Unterschiede:

- Aktuellen Kernel verwenden (3.18.y)
- Neue Konfiguration `bcm2709_defconfig`

Hier die Änderungen:

```
quade@ezs-mobil:~/embedded/raspi$ git clone \
    https://github.com/raspberrypi/linux.git
quade@ezs-mobil:~/embedded/raspi$ cd linux
quade@ezs-mobil:~/embedded/raspi/linux$ git branch -a
* rpi-3.18.y
remotes/origin/HEAD -> origin/rpi-3.18.y
remotes/origin/master
remotes/origin/rpi-3.10.y
remotes/origin/rpi-3.10.y-next
remotes/origin/rpi-3.11.y
remotes/origin/rpi-3.12.y
remotes/origin/rpi-3.13.y
remotes/origin/rpi-3.13.y-next
remotes/origin/rpi-3.14.y
remotes/origin/rpi-3.15.y
remotes/origin/rpi-3.16.y
remotes/origin/rpi-3.17.y
remotes/origin/rpi-3.18.y
remotes/origin/rpi-3.18.y-rebase
remotes/origin/rpi-3.19.y
remotes/origin/rpi-3.2.27
remotes/origin/rpi-3.6.y
```

```

remotes/origin/rpi-3.8.y
remotes/origin/rpi-3.9.y
remotes/origin/rpi-patches
quade@ezs-mobil:~/embedded/raspi/linux$ # Falls Version älter als rpi-3.18.y
quade@ezs-mobil:~/embedded/raspi/linux$ # git checkout rpi-3.18.y
quade@ezs-mobil:~/embedded/raspi/linux$ make ARCH=arm bcm2709_defconfig
quade@ezs-mobil:~/embedded/raspi/linux$ make ARCH=arm \
    CROSS_COMPILE=arm-linux-gnueabi- -j4
...
quade@ezs-mobil:~/embedded/raspi/linux$

```

###Seite 72 Bootpartition

Unten auf der Seite:

Sie müssen jetzt den selbstkompilierten Kernel auf die Bootpartition kopieren. Sie können diesen im Fall eines Raspberry Pi der ersten Generation (A, B, B+) unter dem Standardnamen ``kernel.img``, für einen Raspberry Pi der zweiten Generation (B2) unter dem Standardnamen `*kernel7.img*` ablegen oder aber auch einen individuellen Namen, beispielsweise ``zImage``, wählen. Falls Sie den individuellen Namen vorziehen, benötigen Sie zusätzlich eine Datei ``config.txt``, in der Sie die Zeile `"`kernel=zImage`"` eintragen. Nach dem Kopieren des Kernels kann die Bootpartition wieder ausgehängt werden:

```

quade@felicia:~/embedded/raspi> # Falls vorhanden Original-Kernel sichern
quade@felicia:~/embedded/raspi> mv \
    /media/boot/kernel.img /media/boot/kernel.img.org
quade@felicia:~/embedded/raspi> mv \
    /media/boot/kernel7.img /media/boot/kernel7.org

quade@felicia:~/embedded/raspi> cp \
    linux/arch/arm/boot/zImage /media/boot/kernel.img

```

oder für Raspberry Pi B2

```

quade@felicia:~/embedded/raspi> cp \
    linux/arch/arm/boot/zImage /media/boot/kernel7.img

```

Alternativ (falls der Kernel einen anderen Namen tragen soll):

```

quade@felicia:~/embedded/raspi> cp \
    linux/arch/arm/boot/zImage /media/boot/
quade@felicia:~/embedded/raspi> echo "kernel=zImage" \
    >>/media/boot/config.txt

quade@felicia:~/embedded/raspi> sudo umount /media/boot

```

###Seite 78, 3.4 Der Bootloader *Das U-Boot*

Der Boot-Loader `*Das U-Boot*` steht zur Zeit für den Raspberry Pi B2 nur mit eingeschränkter Funktionalität (kein Ethernet) zur Verfügung. Zwar funktioniert der Zugriff auf die SD-Karte, jedoch bereitet das Laden und Ausführen eines Kernels noch Probleme. Folgende Unterschiede gibt es zum Buch:

- Originalquellcode von `<git://git.denx.de/u-boot.git>` statt der Variante `<git://github.com/gonzoua/u-boot-pi>`.
- Neue Default-Konfiguration ``rpi_2_defconfig`` anstelle von ``rpi_b``.

- Das erzeugte Binary kann direkt verwendet und muss nicht mehr mit einem Header versehen werden.

Damit ergeben sich die folgenden Kommandos (Sie müssen im Folgenden `quade` durch Ihren Usernamen ersetzen beziehungsweise die Pfade anpassen):

```
quade@ezs-mobil:~/embedded/raspi/userland/bootloader$ git clone git://git.denx.de/u-boot-
git
quade@ezs-mobil:~/embedded/raspi/userland/bootloader$ cd u-boot

quade@ezs-mobil:~/embedded/raspi/userland/bootloader/u-boot$ export
CROSS_COMPILE=arm-linux-
quade@ezs-mobil:~/embedded/raspi/userland/bootloader/u-boot$ \
PATH=$PATH:/home/quade/embedded/raspi/buildroot-2013.05/output/host/usr/bin/
quade@ezs-mobil:~/embedded/raspi/userland/bootloader/u-boot$ make rpi_2_defconfig
quade@ezs-mobil:~/embedded/raspi/userland/bootloader/u-boot$ make -j 4
...
```

Die Anpassung des zu ladenden Kernels (Kommando `echo`) muss nur einmalig und nicht mit jedem Neugenerieren des Bootloaders durchgeführt werden. Hier die Installationskommandos:

```
quade@ezs-mobil:~/embedded/raspi/userland/bootloader/u-boot$ cp u-boot.bin
/media/quade/boot/
quade@ezs-mobil:~/embedded/raspi/userland/bootloader/u-boot$ echo "kernel=u-boot.bin"
>>/media/quade/boot/config.txt
quade@ezs-mobil:~/embedded/raspi/userland/bootloader/u-boot$ sudo umount
/media/quade/boot
```

###Seite 104, 4.2.2 Netzwerk-Boot per U-Boot

Da *Das U-Boot* noch Schwierigkeiten macht, ist dieser Abschnitt mit dem Raspberry Pi B2 nicht durchführbar.

###Seite 177, 180 und 188, 6. Gerätetreiber selbst gemacht

Wichtig: Kernel und Kernelquellen müssen zusammenpassen. Das setzt voraus, dass ein eigener Kernel generiert wurde.

Bei der aktuellen Raspbian-Version ist die Majornummer 248 bereits belegt. Daher muss eine andere gewählt werden, beispielsweise 246. Vor dem letzten Absatz auf Seite 177 ist zu ändern:

```
> Funktion `mod_init()` so anpassen, dass die feste Gerätenummer
> `MKDEV(246,0)` verwendet wird.
```

Das betrifft dann auch den Quellcode auf den Seiten 180 und 188 (248 durch 246 ersetzen):

```
static dev_t fastgpio_dev_number = MKDEV(246,0);
```

```
<!--
pandoc Info.md -o Info.pdf -s -S -V geometry=a4paper,left=3cm,right=2cm,top=2cm,bottom=2cm
-->
```